NTB

Interstaatliche Hochschule für Technik Buchs  
Systemtechnikprojekt 2018  
Team 10, Campus Chur

KLaus

Team 10

Inhalt:

[Kurzfassung 2](#_Toc511743163)

[Einleitung 3](#_Toc511743164)

[Platzhalter Teamvorstellung/Struktur 4](#_Toc511743165)

[Konzeptentscheid 5](#_Toc511743166)

[Teilfunktionen 6](#_Toc511743167)

[Übersicht 6](#_Toc511743168)

[Mechanik 7](#_Toc511743169)

[Elektronik 8](#_Toc511743170)

[Spannungsversorgung 8](#_Toc511743171)

[MPC 8](#_Toc511743172)

[Sensorik 8](#_Toc511743173)

[Wlan 8](#_Toc511743174)

[Schalter 8](#_Toc511743175)

[Probleme und Lösungen 8](#_Toc511743176)

[Informatik 9](#_Toc511743177)

[Programmablauf 9](#_Toc511743178)

[Kommunikationskonzept 9](#_Toc511743179)

[Programmaufbau 9](#_Toc511743180)

[Schnittstellen 9](#_Toc511743181)

[Probleme und Lösungen 9](#_Toc511743182)

[Schlussfolgerung 10](#_Toc511743183)

# Kurzfassung

Uns wurde die einmalige Gelegenheit geboten, in den ersten beiden Semestern des Studiums einen Roboter zu bauen. Aus allen Standorten der NTB wurden Teams gebildet, deren Auftrag es war einen Roboter zu bauen, um dann mit einem Partnerroboter aus einem anderen Team einen Turm zu bauen.

Die Bausteine des Turms sind Duplos, welche der Roboter aus den Spendern vorne auf dem Spielfeld entnimmt und dann aufeinanderstapelt, um die Basis eines Leuchtturms zu bilden.

Unserer Roboter heisst Klaus und besteht aus unterschiedlichen Systemen die miteinander interagieren.

# Einleitung

Als Team 10 aus Chur haben wir im Rahmen des Systemtechnikprojektes 2017/2018 die Aufgabe, einen Roboter zu entwickeln und mit Hilfe dessen eine interdisziplinäre, technische Problemstellung zu lösen. Es soll ein Leuchtturm aufgebaut werden, wozu 2 kooperierende Roboter genutzt werden. Team 2 und Team 3 in Buchs entwickeln je einen Partnerroboter. Das Zeitlimit, bis der Turm eigenständig in der geforderten Höhe und mit Leuchtturmspitze steht, beträgt drei Minuten. Zu erkennen ist die Dauer der zur Verfügung gestellten Zeit daran, dass die Turmspitze zu Beginn kurz aufblinkt und am Ende zu leuchten beginnt. Es muss jeweils eine Kommunikation zwischen den beiden Partnerrobotern stattfinden, um die Arbeitsschritte der autonom arbeitenden Roboter aufeinander abzustimmen. Jedem Team steht ein Budget in der Höhe von CHF 800.- zur Verfügung. Das Budget kann durch private Mittel bzw. allfällige Sponsoren erhöht werden. Das Systemtechnikprojekt unterliegt einem festgelegten Zeitfenster. Es müssen verschiedene Meilensteine eingehalten werden. Bis zum 26.06.2018 muss das Projekt endgültig abgeschlossen sein und einer Fachjury, sowie dem öffentlichen Publikum präsentiert werden. Um den Erfolg zu gewährleisten muss das Wort «Team» grossgeschrieben werden. Doch eine gute Zusammenarbeit gelingt nur, wenn jedes einzelne Teammitglied Eigenverantwortung und Eigeninitiative wahrnimmt.

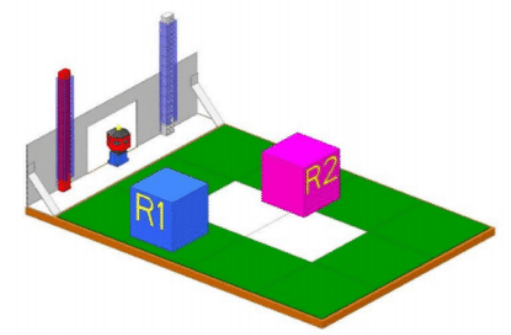


Abbildung Spielfeld

## Platzhalter Teamvorstellung/Struktur

Entwicklungsverlauf des Konzepts.

* Ideensammlungsphase
* Skizzenphase (Bilder)
* Aufteilen der Teilaufgaben
* Beginn der Ideenanalyse auf Möglichkeiten und Durchführung
* Überarbeiten der Ideen -> neue Ideenfindung
* Endgültige übereinstellung der Konzepte
* Konzeptentscheid

# Konzeptentscheid

Für die Konzeptfindung wurde im Vorfeld ein Brainstorming gemacht und daraus ein morphologischer Kasten erstellt:



Nach genauer Abschätzung aller Varianten und Gewichtung von dessen Vor- und Nachteile wurde ein Konzept festgelegt.

|  |  |
| --- | --- |
| Energieversorgung | Lithium-Polymer-Akkumulator |
| Kommunikation | Wifi Modul |
| Fortbewegung | 4 Räder, die in den Rillen der Legoplatte fahren |
| Antrieb | Bürstenloser Gleichstrommotor mit Encoder und Servo-Motor |
| Greifmechanismus | Magnetgreifkopf |
| Turm bauen | «Gabelstapler», ausfahrbarer Arm |
| Orientierung | IR-Sensoren und Encoder |

Für die Entscheidung des Konzeptes war die Zusammenarbeit mit den beiden Partnerteams von Buchs, Team 2 und 3, von grosser Bedeutung. Nach intensiven Diskussionen wurde eine endgültige Entscheidung gefällt, die folgenden Inhaltspunkte behält:

-Team 10 operiert auf der linken Spielfeldhälfte, Team 2 /3 auf der rechten.

-Team 10 legt den ersten Stein.

-Danach werden abwechselnd Steine gelegt, bis die erforderte Höhe erreicht ist.

-Team 2/3 setzt die Turmspitze.

# Teilfunktionen

## Übersicht

Img -> Bild Funktionsübersicht -> Bewegungen des Roboters

//Auf Details wird in den nächsten Kapitel eingegangen.

## Mechanik

## Fortbewegung

### Unterteil

#### Bodenplatte

Auf der Bodenplatte des Fahrwerks wurden der Motor für die Fortbewegung, die Achsenlager für die Räder, sowie der Drehstützen für die Verbindung zwischen Fahrwerk und Oberteil befestigt. Ausserdem wurde an der Bodenplatte ein Lichtsensor befestigt, welcher sicherstellt, dass der Roboter vor dem Ablegen des Steines am richtigen Ort ist.

#### Deckplatte

Auf der Deckplatte des Fahrwerks wurde eine gut gleitende Platte aufgebracht, um eine reibungsarme und ebene Oberfläche für die Drehung des Oberteils zu schaffen.

### Oberteil

#### Drehplatte

### Auf der Bodenplatte des Oberteils befinden sich der Antriebsmotor für die Drehbewegung, der Antriebsmotor für die horizontale Bewegung des Armes, welche über ein Riemenband auf den Arm übertragen wird, sowie das Gegenstück zum Drehstütze.

#### Zwischenplatte

Auf der Zwischenplatte werden die elektronik-Platinen befestigt.

#### Deckplatte

Auf der Deckplatte des Oberteils wird die Rampe befestigt, welche für das Aufklappen des Klapparms zuständig ist. Ausserdem ermöglichen Aussparungen in der Deckplatte einen erleichterten Zugang auf die darunterliegende Elektronik.

#### Greifarm

Der Arm wird an einer vertikalen Gleitschiene befestigt, welche wiederum an zwei horizontalen Gleitschienen befestigt sind, welche an der Boden- und Deckplatte des Oberteils angebracht werden. Er besteht aus dem Arm und dem Greifer. Der Arm ist das Verbindungsstück zwischen Schiene und einem Servo-Motor. Dieser erlaubt es, den an ihm angebrachten Greifer um 90° zu drehen.

### Vorgehen

Wir wussten seit der Spezialisten Woche, dass der Roboter eine Kraft von ungefähr 45 Newton überwinden muss, um den ersten Stein auf das Spielfeld zu drücken. Daher war es uns wichtig, dass der Roboter genug schwer ist, so dass er nicht wegkippt, wenn er versucht den Stein anzudrücken. Daher haben wir bei allen fünf Platten Stahl verwendet, um das nötige Gewicht zu erreichen.

Bei der Dimensionierung der Fortbewegungsmotoren kamen folgende Berechnungen zustande:



Bei der Dimensionierung der anderen Motoren haben wir uns mit dem Informatikteam und dem Elektronikteam besprochen und ausserdem Rat von einem Mitarbeiter von Faulhaber eingeholt.

Dies ergab die folgenden Motoren und Getriebe von Faulhaber:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Anwendung | Motor | Drehmoment | U/min | Getriebe |
| Antrieb | 2232R012SR | 10 mNm | 5510 min-1 | 173:1 |
| Drehung | 2232R012SR | 10 mNm | 5510 min-1 | 173:1 |
| Horizontale Verschiebung | 1524T012SR | 2.9 mNm | 4130 min-1 | 485:1 |
| Vertikale Verschiebung | 2232R012SR | 10 mNm | 5510 min-1 | 308:1 |

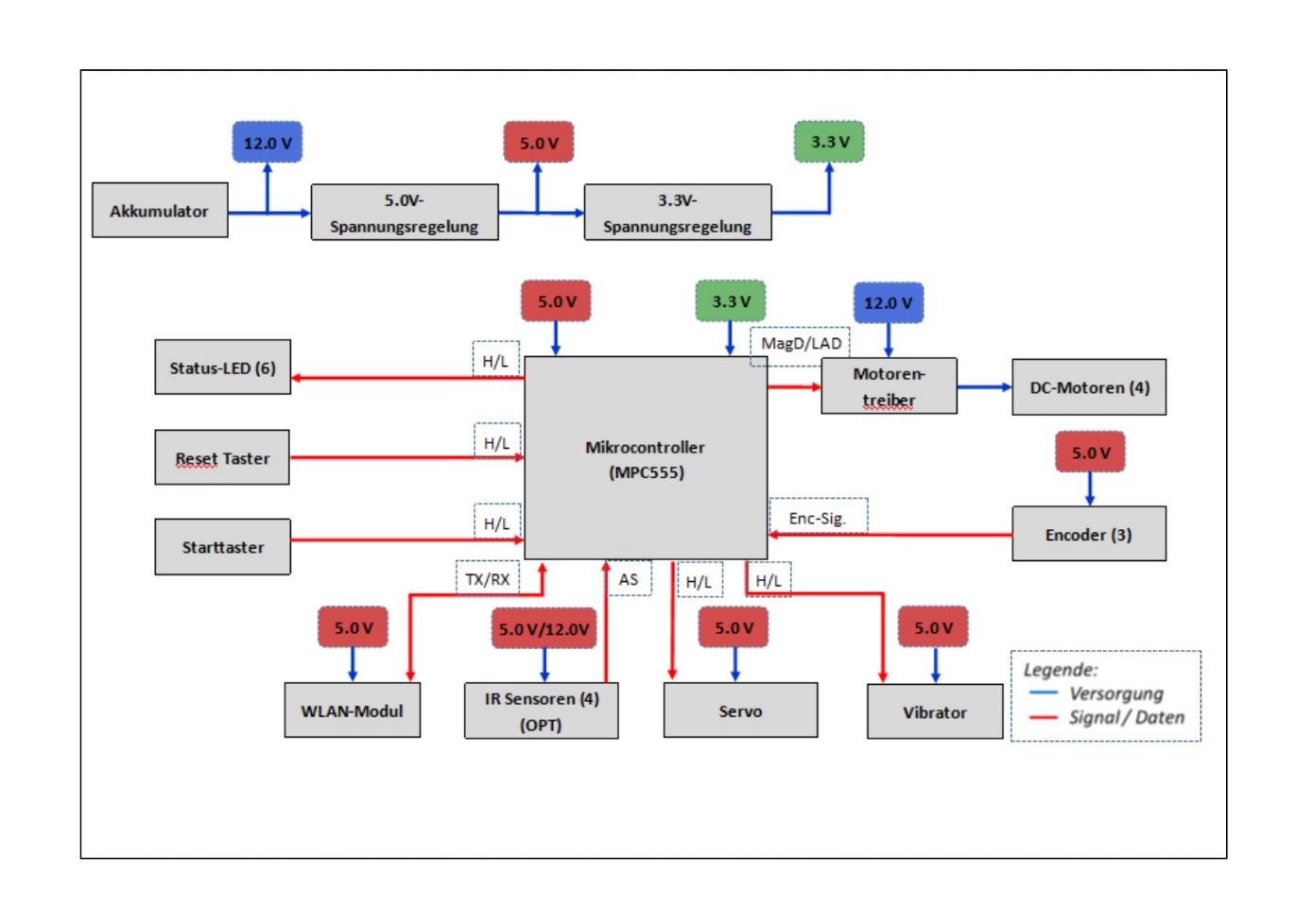
### Probleme/ Lösungen

Eines unserer grössten Probleme von Anfang an war die Bedienung von Creo. Da wir in unserem Team weder einen gelernten Mechaniker noch einen Konstrukteur hatten, fehlten uns entscheidende Erfahrungen für die Konstruktion des Roboters. Als dann noch nach dem ersten Semester ein Mitglied des Team Mechanik das Studium abbrach, wurde die Situation noch schwieriger. Einer unserer « Informatiker » hat sich freiwillig umteilen lassen und erlernte das Zeichnen mit einem CAD Programm so schnell wie möglich.

Wir stellten schnell fest, dass wir möglichst wenig Teile aus Kunststoff herstellen durften und stattdessen Aluminium oder Stahl verwenden mussten. Dadurch erhöht sich die Stabilität und das Risiko eines Materialversagens wird minimiert. Nach dem Meilenstein M3 wurde uns bewusst wir sehr wir mit der Mechanik im Rückstand waren und mussten daher handeln. Wir überarbeiteten unsere Problemstellen nochmals und vereinfachten sie soweit wie möglich. Wir versuchten möglichst viele Teile zu bestellen anstatt selbst zu konstruieren. Als uns klar wurde, dass wir komplett die falschen Getriebe bestellt hatten bestellten wir neue, Faulhaber war so grosszügig uns neue zu liefern. Doch diese waren einiges grösser als die Vorherigen. Dies stellte uns erneut vor Probleme auf Grund von Platzmangel. Als wir dann noch Zahnräder in der falschen Grösse bestellt haben und zusätzlich keine Befestigungsmöglichkeit bot, entschieden wir uns dafür als erstes darauf zu achten, dass die Funktionstüchtigkeit des Roboter gewährleistet werden kann und erst dann um die Abmasse. Da die Zeit drängte haben wir die Teile bei einem unserer Mitglieder zuhause gefertigt, es hätte zu viel Zeit in Anspruch genommen für jedes Detail auf Buchs zu fahren und mit dieser Werkstatt zusammen zu arbeiten.

## Elektronik

Blockschaltbild:



Img -> Die beiden Boards

Das Blockschaltbild ist eine übersichtliche Darstellung für die Aufteilung der Elektronik. In Realität wurden zwei separate Printplatten erstellt.

Die untere Printplatte beinhaltet:

-Spannungsversorgung 12V

-Transformation von 12V zu 5V

-Transformation von 12V zu 3.3V

-2 Motorentreiber Locked Antiphase

-1 Motorentreiber Sign Magnitude

-Signalisations-LED für Spannungsversorgung

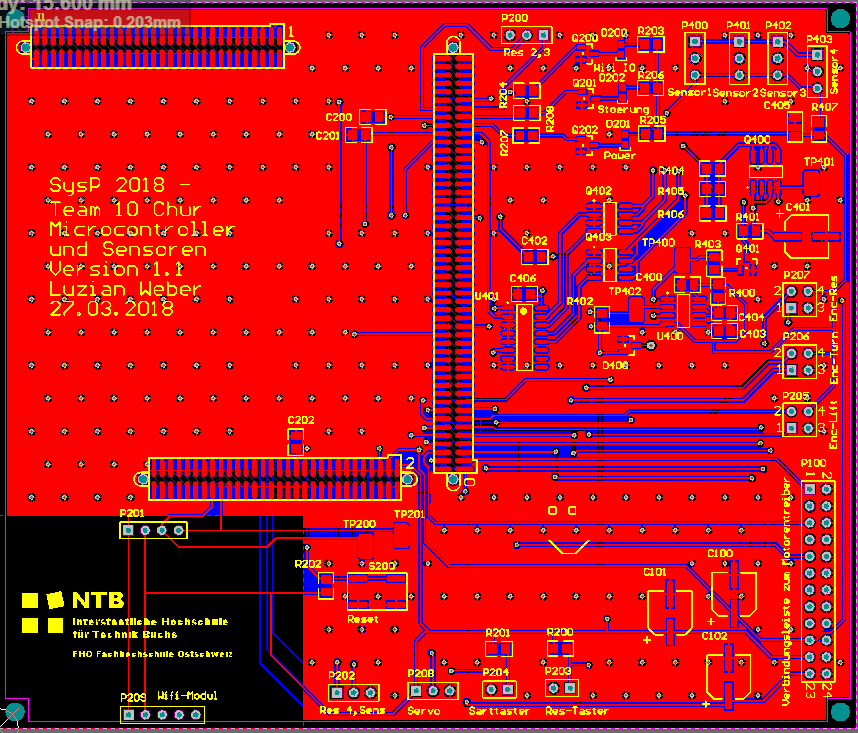
Spannungsversorgung

Die Spannungsversorgung erfolgt über ein Lithium-Polymer-Akkumulator (11.1 V / 1300 mAh)

(Foto)

Detailierte Beschreibung Michael

Bild

Die obere Printplatte ist mit folgenden Bauteilen bestückt:

IR-Sensore

Start- und Reservetaster

Vibrator Anschluss

Wifi-Modul

Encoder Anschlüsse

Signalisations-LED

Anschlussplatte MPC 555

Resett-Taster

### MPC

Der MPC 555 ist das Herzstück und das Gehirn des Roboters. Von ihm aus werden sämtliche Ansteuerungen für Motoren, Sensoren, LEDs, etc. getätigt.

Foto/Schema

### Sensorik

Für die Orientierung des Roboters wurden IR-Sensoren eingesetzt. Der IR-Emitter sendet IR Licht aus, das an einem Objekt reflektiert. Anhand der Intensität des IR Lichtes, welches beim IR Empfängers (Reciever) ankommt, kann auf die Distanz zwischen dem reflektierenden Objekt und IR Sensor geschlossen werden.

Verwendet wurden die HLC 1395 Sensoren, welche ideal für kurze Distanzen eingesetzt werden können. Eine genauere Beschreibung kann dem Datenblatt auf der CD entnommen werden.

Foto

Verwendete Bereiche der Sensoren:

* In Greifer für die Vorwärtsfahrt
* Unten am Roboter für die Rückfahrt

### Encoder

Für die genaue Positionsmessung wurden Encoder eingesetzt. Die Encoder der Serie IEH2-4096 wurden direkt von Faulhaber mit den Motoren mitgeliefert. Eine genauere Beschreibung kann dem Datenblatt auf der CD entnommen werden.

Einsatzbereich der Encoder:

* Positionsmessung des Armes horizontal
* Positionsmessung des Armes vertikal
* Drehmotor

### Wlan

Für die Kommunikation wurde auf die empfohlene Variante der NTB gesetzt, das WLAN Modul. Das Wifi Modul RN-131GXS ist ein fertig bestücktes Bauteil, das von der NTB zu Verfügung gestellt wurde. Für eine optimale Kommunikationsverbindung mit dem Partnerteam wurde das Modul frei von metallischen Umhüllungen platziert. Auch das Polygon auf unserer Printplatte wurde unter dem Modul ausgeschlossen. (Siehe Bild PCB)

### Probleme und Lösungen

Kein experte zu CAD Program im Team

Ursprünglich keine Ahnung von altium design

Falscher footprint hat sich eingeschlichen. -> einlöten von Drähten

IR-Sensoren gaben falsche daten aus. -> Falsche Widerstände auf dem Board.

## Informatik

Kleine Einführung in die Funktion der Informatik

### Programmablauf

Der Roboter kennt insgesamt 10 verschiedene Zustände.

Im Ersten wird die Kommunikation mit dem Partnerroboter aufgebaut und alle Motorentreiber unter Strom gesetzt. Anschliessend wird das Startsignal gesendet und der nächste Zustand wird aktiviert.

Hier fährt Klaus das erste Mal nach vorne und fährt seinen Greifarm aus. Wenn der Sensor im Greifer vorne erkennt das er vorne ist, stoppt der Roboter.

Jetzt starte er einen Timer damit er weiss, wann der den Arm heben darf. Dann fährt er zurück und sobald der Timer abgelaufen ist, fährt auch der Greifarm nach oben.

Wenn nun der hintere Sensor den Legostein auf der linken Seite erkennt (welchen wir vor dem Start dort setzten dürfen), hält der Roboter an und wartet bis das Signal vom Partnerroboter kommt, dass die Baustelle frei ist. Weil wir den ersten Stein setzten, wird beim ersten Mal nicht darauf gewartet und Klaus drehen direkt ab.

Ist der Legostein nun richtig positioniert, fährt der Arm nach unten und drückt den Stein an. Hiernach fährt der Arm wieder nach oben und Klaus dreht sich wieder Richtung Legospender. Gleichzeitig sendet Klaus dem Partnerroboter ein Signal mit der Anzahl gesetzter Legosteinen, was gleichzeitig bedeutet, dass nun er seinen Stein setzten kann.

Ist nun die gewünschte Höhe von neun Legosteinen erreicht. Stoppen wir und senden das erste Stoppsignal. Wenn dies nicht der Fall ist, Fährt der Roboter wieder nach vorne und holt den nächsten Stein, bis die Höhe erreicht ist.

### Kommunikationskonzept

Die Kommunikation wird über Fire&Forget-System gemacht, es wird keine Bestätigung für das erhalten einer Nachricht verschickt. Die einzige Ausnahme dazu ist das Signal 222, das zu Beginn verschickt wird, um zu überprüfen, ob der Partnerroboter bereit ist und die Kommunikation steht. Falls dies nicht der Fall ist, wird das Signal nach kurzer Zeit wiederholt. Sobald die Bestätigung kommt, wird das Startsignal an den Leuchtturm und den Partnerroboter geschickt.

Während des Bauens wird jeweils das Setzen eines Steins mit einer Nachricht an den Partnerroboter bestätigt, um Komplikationen mit der Nummer 10, die beim Testen Probleme verursachte, zu vermeiden, wird jeweils die Höhe in Steinen+100 gesendet. Sobald unser Roboter den letzten Stein gesetzt hat, schickt er das Schlusssignal an den Leuchtturm.

### Programmaufbau\\fs003\stud\rjaeger\Desktop\SysP18_1cv\04_Informatik\01_Diagramme\Class_diagram_2.png

Die Klasse Robi ist die Klasse, von der aus der Ablauf gesteuert wird. Alle Funktionen anderer Klassen werden von dieser Klasse aus gesteuert. Die verschiedenen Zustände sind im Abschnitt Zustandsdiagramm genauer erläutert.

Die Klasse Timer beinhaltet eine Stoppuhr, durch welche man Zeitabstände, die nicht durch Tasks gesteuert werden, messen kann.

Die Kommunikation wird durch die Klasse Wifi gesteuert. Der genaue Ablauf der Kommunikation sind im Abschnitt Kommunikationsdiagramm erklärt.

Die Klasse IO instanziiert Ein- und Ausgänge, in der Klasse PinMap sind alle benutzten Pins aufgeführt. Diese Klassen helfen, Ordnung zu halten, da man immer weiss, wo ein Pin zu finden ist und man mit einem Klick den Wert für alle Aufrufe dieses Pins verändern kann. In RobiConstants werden Konstanten gespeichert, um diese mit einem Klick verändern zu können.

Sensoren beinhaltet eine Liste der DistSensoren, diese beiden Klassen verwalten die Distanzsensoren und beinhalten Methoden um Hindernisse zu erkennen oder zu testen, ob ein bestimmter Grenzwert erreicht wurde.

Servo steuert den Servomotor, für dieses Projekt wurden nur die Maximal- und Minimalposition gebraucht.

Die Klasse Encoder speichert die Position eines Motors. Ein Encoder kann mit einem Faktor initialisiert werden, damit alle Übersetzungen unterstütz werden.

Drive ist eine abstrakte Klasse, die entweder vom Typ LockedAnti (=Locked Antiphase) oder SignMagn (=Sign Magnitude) ist. Die verschiedenen Ausprägungen müssen die Methode setSpeed überschreiben, damit man beide Arten von Motoren gleich ansteuern kann. Ausserdem wird beim Instanziieren eines Motors festgelegt, ob dieser über einen Encoder oder über Sensoren gesteuert wird.

Die Klasse Move verwaltet die Bewegung des Roboters und steuert zwei Sensoren sowie ein Locked-Antiphase-Motor ohne Encoder für die Bewegung auf dem Spielfeld und ein Sign-Magnitude-Motor mit Encoder für die Drehung des Roboters.

Die Klasse Lift ist ähnlich wie die Klasse Move, steuert aber den Greifmechanismus.

Beide Klassen stellen Methoden für die Klasse Robi zur Verfügung, um diese Klasse möglichst übersichtlich zu halten.

### Schnittstellen

Was sind schnittstellen…

#### Ansteuerung von Motoren

Img -> codeschnipsel aus ansteuerung motor

Beschrieb codeschnipsel

#### Grenzwerte für Sensoren

Img -> codeauschnitt Grenzwerte

Beschrieb codeschnipsel

### Probleme und Lösungen

Die erste Version des Programmes hatte nur wenige Klassen, was das Testen und die Fehlersuche sehr schwer gestalten würde. Ausserdem sind die einzelnen Klassen so sehr unübersichtlich. Auf einen Ratschlag des damaligen Betreuers, Ulrich Hauser, wurde das Programm neu strukturiert. Die oben genannten Diagramme basieren auf diesem Stand.

Von den Studenten, die das Projekt bereits hinter sich hatten, wurde geraten, Klassen wie PinMap einzubauen, um Konstanten an einem Ort zu zentrieren, was Änderungen vereinfacht.

Während des Testens gab es ein Problem mit den Locked Antiphase Motoren. Das Problem war, dass die Initialisierung an der falschen Stelle im Programm stattfand.

Zu Beginn machte das WLAN-Modul Probleme. Diese sind auf Konflikte mit der Task-Superklasse zurückzuführen. Als diese Probleme gelöst waren, funktionierte es immer noch nicht, auf dem Testboard funktionierte es aber. Somit wurde das WLAN-Modul ausgetauscht, mit dem Neuen funktioniert es.

# Schlussfolgerung

Was haben wir gelernt

Was nehmen wir mit für die Zukunft

Wie fühlen wir uns als Team (Hochs und Tiefs)